An improved branch-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem

Guillaume Marques, Ruslan Sadykov, Jean-Christophe Deschamps, Rémy Dupas

An improved branch-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem

Guillaume Marques*[1,2,3], Ruslan Sadykov†[2,3], Jean-Christophe Deschamps‡[1], and Rémy Dupas§[1]

[1]IMS, Univ. of Bordeaux, Bordeaux INP, CNRS (UMR 5218) , 351 Cours de la libération, 33405 Talence cedex, France

[2]Inria Bordeaux – Sud-Ouest , 200 Avenue de la Vieille Tour, 33405 Talence cedex, France

[3]IMB, Univ. of Bordeaux, CNRS (UMR 5251), Bordeaux INP, 351 Cours de la libération, 33405 Talence cedex, France

23 September 2019

Abstract

In the paper, we propose a branch-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem in which delivery of products from a depot to customers is performed using intermediate depots called satellites. Our algorithm incorporates significant improvements recently proposed in the literature for the standard capacitated vehicle routing problem such as bucket graph based labeling algorithm for the pricing problem, automatic stabilization, limited memory rank-1 cuts, and strong branching. In addition, we make some specific problem contributions. First, we introduce a new route based formulation for the problem which does not use variables to determine product flows in satellites. Second, we introduce a new branching strategy which significantly decreases the size of the branch-and-bound tree. Third, we introduce a new family of satellite supply inequalities, and we empirically show that it improves the quality of the dual bound at the root node of the branch-and-bound tree. Finally, extensive numerical experiments reveal that our algorithm can solve to optimality all literature instances with up to 200 customers and 10 satellites for the first time and thus double the size of instances which could be solved to optimality.

1 Introduction

In a context of economic globalization, the growth of urban population leads to an increase in freight transportation in cities. Freight transportation may deteriorate the quality of the urban environment with, for example, high noise levels, greenhouse gas emissions, and decrease of air quality. Moreover, freight transportation faces several constraints such as reduced access within cities and deliveries within time-windows. As a result, freight distribution patterns in city logistics change (Taniguchi and Thompson, 2002). In the past, customers were delivered straight from depots located on the outskirts of cities. Nowadays, transporters tend to use two-tier distribution systems. In the first tier, large urban trucks ships freight from warehouses

*guillaume.marques@inria.fr
†ruslan.sadykov@inria.fr
‡jean-christophe.deschamps@ims-bordeaux.fr
§remy.dupas@ims-bordeaux.fr

or production sites to intermediate distribution facilities called satellites. In satellites, freight is processed and consolidated. Freight is then loaded in small city freighters which deliver customers located in city centers.

At the strategic level, two-tier distribution systems are considered in location-routing problems (Crainic et al., 2011). These are integrated problems in which we take decisions on both locating facilities and routing from open facilities. At tactical and operational levels, locations of depots and satellites are known. We plan only routing of vehicles. However, we should take routing decisions on both levels at the same time to devise cost-effective solutions in two-tier distribution systems. Such integration gave rise to two-echelon routing problems (Crainic et al., 2009). The first such problem proposed in the literature by Gonzalez-Feliu et al. (2007) is the two-echelon capacitated vehicle routing problem (2E-CVRP).

In the 2E-CVRP, we must determine the number of goods to be shipped from the depot to the satellites and from satellites to customers, and the optimal routes connecting entities in each level such that vehicle capacities are not exceeded. We aim at minimizing the sum of handling costs at satellites and transportation costs depending on the total distance traveled by all vehicles.

Recently, several exact algorithms for the 2E-CVRP have been proposed in the literature. The most efficient one by Baldacci et al. (2013) is based on an enumeration of collections of first-level routes. Thus, it can efficiently tackle only instances with a small number of satellites (up to six). Moreover, this algorithm solves to optimality instances with up to 100 customers whereas the best exact algorithms for other vehicle routing problems can handle up to 300 customers (Pecin et al., 2017a). Santos et al. (2015) proposed the only branch-cut-and-price algorithm in the literature for the 2E-CVRP. This algorithm can be used to solve instances with a larger number of satellites, but experimentations show that it is less efficient than the one of Baldacci et al. (2013).

In this paper, we propose an improved branch-cut-and-price (BCP) algorithm for the 2E-CVRP which is built on recent advances for the classical capacitated vehicle routing problem (CVRP). To further improve the efficiency of our BCP algorithm, we propose the following problem-specific enhancements:

- A new route based formulation for the problem which does not involve variables which explicitly define product flow in satellites. New level balancing constraints guarantee flow conservation in satellites.
- A new family of inequalities to improve the quality of the linear programming (LP) relaxation of the formulation. These inequalities are inspired by the depot capacity constraints introduced for the capacitated location-routing problem by Belenguer et al. (2011).
- A new branching strategy which uses variables defining the number of urban trucks visiting a subset of satellites.

To improve the current primal bound, we employed a column generation based heuristic in the course of the algorithm. Our BCP algorithm with the embedded heuristic outperformed largely the state-of-the-art exact approach by Baldacci et al. (2013) for the problem, since it solved to optimality all instances available in the literature with up to 200 customers and 10 satellites.

Finally, we generated a new set of large instances for the problem to inspire further research on the 2E-CVRP. This set involves instances with up to 300 customers and 15 satellites. These instances are derived from ones recently proposed by Schneider and Löffler (2019) for the capacitated location-routing problem.

The remaining of the paper is organized as follows. Section 2 reviews the literature. Section 3 describes the standard and the new formulations of the problem. Section 4 introduces the new family of satellite supply inequalities. Section 5 describes the proposed branch-cut-and-price algorithm. Section 6 reveals and discusses the computational results. Section 7 concludes and presents further research perspectives.

2 Literature review

Gonzalez-Feliu et al. (2007) first considered the 2E-CVRP. They proposed a freight-flow formulation enhanced by two families of valid inequalities. Their branch-and-cut algorithm solved to optimality instances with up to 22 customers and 2 satellites. Perboli et al. (2011) improved these results. The authors strengthened the formulation with one family of valid inequalities. They solved to optimality some instances with 33 customers. Two matheuristics were also suggested. They found feasible solutions to instances up to 50 customers with 10% of average gap from the lower bound.

Later, Jepsen et al. (2013) pointed out that the formulation in (Perboli et al., 2011) is not correct for instances with more than two satellites. They proposed an alternative formulation that combines the relaxation of the split-delivery CVRP by Belenguer et al. (2000) for the first level and the model for the capacitated location routing problem by Contardo et al. (2013) for the second level. Although this formulation is an outer approximation, its LP relaxation is stronger than one of Perboli et al. (2011). Since this formulation has an exponential number of constraints, the authors used a branch-and-cut algorithm. A specialized branching scheme was employed to cut non-feasible integer solutions. This approach solved to optimality instances with up to 50 customers and 5 satellites. It remains the best branch-and-cut algorithm for the problem in the literature.

Contardo et al. (2012) proposed another branch-and-cut algorithm for the two-echelon capacitated location-routing problem. This problem is a generalization of the 2E-CVRP in which there are several depots and opening costs for satellites. Their branch-and-cut algorithm solved to optimality instances with up to 50 customers and 10 potential satellites.

Santos et al. (2015) proposed the first branch-cut-and-price algorithm for the 2E-CVRP. They considered a route based formulation strengthened by some valid inequalities from the CVRP literature. First-level routes are enumerated whereas second-level routes are priced by the shortest path problem with resource constraints. The pricing problem generates non-elementarity routes. They used branching strategies in the following priority order: (1) branching on the number of vehicles traveling on a first-level route, (2) branching on the number of second-level vehicles starting from a satellite, and (3) branching on the use of an arc by a second-level route. The computational results of Santos et al. (2015) were similar to those by Jepsen et al. (2013).

The exact approach by Baldacci et al. (2013) also uses a route based formulation. The method is based on an intelligent enumeration of collections (i.e. subsets) of first-level routes. Authors devised lower and upper bounding procedures to limit the number of collections which may lead to an optimal solution. By fixing a subset of first-level routes, the problem is reduced to the multi-depot capacitated vehicle routing problem with limited depot capacities. The latter was solved by an adaptation of the algorithm by Baldacci and Mingozzi (2009). Computational experiments showed that the overall approach outperforms the one by Jepsen et al. (2013). Baldacci et al. (2013) could solve instances with up to 100 customers and 5 satellites. Their approach remains the best exact algorithm in the literature until now. However, the fact that collections of first-level routes are enumerated does not allow one to employ this approach for instances with 10 satellites or more.

There are several heuristic approaches for the 2E-CVRP in the literature. Hemmelmayr et al. (2012) proposed an adaptive large neighborhood search based heuristic that works for both 2E-CVRP and the location-routing problem. It largely improved the best feasible solutions found by Perboli et al. (2011). Moreover, the authors proposed a new test set of instances with up to 200 customers and 10 satellites.

Zeng et al. (2014) suggested a hybrid heuristic which is composed of a greedy randomized adaptive search procedure (GRASP) with a route-first cluster-second procedure embedded in a variable neighborhood descent (VND). Breunig et al. (2016) developed an improved large neighborhood-based hybrid meta-heuristic. It combines enumerative local search with destroy-and-repair principles, as well as some tailored operators to optimize the selections of satellites. Both these approaches improved the best-known solutions for many instances.

Recently, two matheuristics were proposed in the literature. Wang et al. (2017) employed a mixed-integer mathematical model for the 2E-CVRP, in which arc variables are used for the first level, and path variables for the second level. They used variable neighborhood search to construct the set of second-level routes, and they then solved the mathematical model to improve the obtained solution. The authors improved 13 best-known solutions. Finally, Amarouche et al. (2018) used a similar approach in which a pool of routes is collected by a local search heuristic combined with a destroy-and-repair method. Then, the route based formulation is solved with the hope to improve the best solution found so far. This approach improves 7 best-known solutions for the largest instances of the 2E-CVRP.

Further information can be found in the survey on two-echelon routing problems by Cuda et al. (2015).

3 Formulation

Let us now formally define the problem. At the first level, a set $\mathcal{K}$ of homogeneous urban trucks ships freight from a depot denoted as 0 to a set $\mathcal{S}$ of intermediate depots called satellites. At the second level, a set $\mathcal{L}$ of homogeneous city freighters picks freight at satellites and deliver it to a set $\mathcal{C}$ of customers. Each vehicle must return to the place from where it started its tour (depot for urban trucks and satellites for city freighters). Urban trucks have a capacity of Q_1 items, and city freighters have a capacity of Q_2 items. A satellite $s \in \mathcal{S}$ can hold up to L_s city freighters and charges f_s^H for each processed item of freight. Each customer $c \in \mathcal{C}$ asks for d_c items of freight and must be visited exactly once. At each satellite, the total amount of freight delivered by urban trucks must be equal to the amount of freight picked by city freighters that start at this satellite. The objective of the problem is to minimize the sum of handling costs f^H and transportation costs f^T.

We use the route based formulation to model the problem. The first level is similar to the split-delivery CVRP since several urban trucks can supply a satellite. However, the amount of freight delivered to each satellite is not known. A complete undirected graph $G_1 = (V_1, E_1)$, $V_1 = \{0\} \cup \mathcal{S}$, represents the first level of the distribution system. Let P be the set of feasible first-level routes and let $\tilde{z}_e^p \in \mathbb{N}$ denote the number of times path $p \in P$ uses edge $e \in E_1$.

The second level corresponds to the multi-depot CVRP where depots are satellites. Each customer is visited by one city freighter, and each satellite cannot supply more freight than the amount delivered to it by urban trucks. This level is represented by an undirected graph $G_2 = (V_2, E_2)$ where $V_2 = \mathcal{C} \cup \mathcal{S}$ and $E_2 = \{(i,j) \mid i \in \mathcal{S} \cup \mathcal{C}, j \in \mathcal{C}, i \neq j\}$. For any satellite $s \in \mathcal{S}$, let R_s be the set of feasible second-level routes starting and finishing in s. We denote $R = \cup_{s \in \mathcal{S}} R_s$. A second-level route $r \in R$ is described by vector $\tilde{x}^r$ where element $\tilde{x}_e^r \in \mathbb{N}$ denotes the number of times route r uses edge $e \in E_2$. We also introduce vector $\tilde{y}^r$ where element $\tilde{y}_c^r \in \mathbb{N}$ denotes the number of times route r visits customer $c \in \mathcal{C}$. Given graph G_i, $i = 1, 2$, we denote as $\delta_i(v)$ the set of edges in E_i incident to vertex $v \in V_i$.

The cost of traversing edge $e \in E_1 \cup E_2$ is denoted by f_e^T. From now on, we make two assumptions. First, transportation costs f^T satisfy the triangle inequality. Otherwise, we transform the instance to an equivalent one: if the minimum cost path between two vertices v, v' passes by other vertices, we set the cost $f_{(v,v')}^T$ to the cost of the minimum path. The second assumption is that transportation costs are symmetric. If this is not the case, graphs G_1 and G_2 become directed ones, edges become arcs, and all values $\tilde{z}_{e=(v,v')}$ and $\tilde{x}_{e=(v,v')}$ depending on edges are replaced by sums of values $\tilde{z}_{a=(v,v')} + \tilde{z}_{a=(v',v)}$ and $\tilde{x}_{a=(v,v')} + \tilde{x}_{a=(v',v)}$ depending on arcs. All instances in the 2E-CVRP literature satisfy these two assumptions.

3.1 Standard formulation

We now describe the standard route based formulation for the 2E-CVRP. It was proposed by Baldacci et al. (2013) and used in Santos et al. (2015); Amarouche et al. (2018). Integer variable

λ_p is equal to the number of urban trucks traveling on first-level route $p \in P$. We denote as S_p the set of satellites visited by route $p \in P$: $S_p = \{s \in \mathcal{S} : \sum_{e \in \delta_1(s)} \tilde{z}_e^p = 2\}$. We denote as P_S the set of first-level routes visiting at least one satellite in $S \subseteq \mathcal{S}$: $P_S = \{p \in P : S_p \cap S \neq \emptyset\}$. Continuous variable w_{ps} is equal to the amount of freight that first-level route $p \in P_{\{s\}}$ delivers to satellite $s \in S_p$. Binary variable μ_r is equal to one if and only if a city freighter travels on second-level route $r \in R$. To simplify the presentation, we introduce the continuous auxiliary variable b_s that is equal to the total amount of freight delivered to satellite $s \in \mathcal{S}$.

$$
\begin{aligned}
(F1) \quad \min \quad & \sum_{p \in P} \sum_{e \in E_1} f_e^T \tilde{z}_e^p \lambda_p + \sum_{r \in R} \sum_{e \in E_2} f_e^T \tilde{x}_e^r \mu_r + \sum_{s \in \mathcal{S}} f_s^H b_s && && (1) \\
\text{s.t.} \quad & \sum_{r \in R} \sum_{c \in \mathcal{C}} \tilde{y}_c^r \mu_r = 1 && c \in \mathcal{C} && (2) \\
& \sum_{r \in R_s} \mu_r \leq L_s && s \in \mathcal{S} && (3) \\
& \sum_{r \in R} \mu_r \leq |\mathcal{L}| && && (4) \\
& \sum_{p \in P} \lambda_p \leq |\mathcal{K}| && && (5) \\
& \sum_{s \in S_p} w_{ps} \leq Q_1 \lambda_p && p \in P && (6) \\
& b_s = \sum_{p \in P_{\{s\}}} w_{ps} && s \in \mathcal{S} && (7) \\
& b_s = \sum_{r \in R_s} \sum_{c \in \mathcal{C}} d_c \tilde{y}_c^r \mu_r && s \in \mathcal{S} && (8) \\
& \lambda_p \in \mathbb{N} && p \in P && (9) \\
& \mu_r \in \{0, 1\} && r \in R && (10) \\
& w_{ps} \geq 0 && p \in P, s \in S_p && (11)
\end{aligned}
$$

Objective function (1) minimizes the sum of transportation and handling costs. Constraints (2) ensures each customer is visited by exactly one second-level route. Constraints (3), (4), and (5) are upper bounds on the number of used vehicles. Constraints (6) make sure that the capacity of urban trucks is not exceeded. Constraints (7) and (8) ensure the flow balance between the two distribution levels. Constraints (9), (10) and (11) define domains of variables.

3.2 Modified formulation

In F1, we use variable w together with constraints (6) and (7) to ensure the flow balance between two distribution levels. To obtain the modified formulation, we project F1 to the space of variables λ and μ only. Investigation of this kind of projection for the CVRP has been performed by Letchford and Salazar-González (2006). To simplify the presentation, we introduce auxiliary integer variables u_S that define the number of urban trucks visiting a non-empty subset $S \subseteq \mathcal{S}$ of satellites. The modified formulation is then

$$
\begin{aligned}
(F2) \quad \min \quad & \sum_{p \in P} \sum_{e \in E_1} f_e^T \tilde{z}_e^p \lambda_p + \sum_{r \in R} \sum_{e \in E_2} f_e^T \tilde{x}_e^r \mu_r + \sum_{s \in \mathcal{S}} f_s^H b_s \\
\text{s.t.} \quad & \sum_{r \in R} \sum_{c \in \mathcal{C}} \tilde{y}_c^r \mu_r = 1 && c \in \mathcal{C} \\
& \sum_{r \in R_s} \mu_r \leq L_s && s \in \mathcal{S}
\end{aligned}
$$

$$\begin{aligned}
&\sum_{r\in R}\mu_r \le |\mathcal{L}| && \\
&\sum_{p\in P}\lambda_p \le |\mathcal{K}| && \\
&\quad b_s = \sum_{r\in R_s}\sum_{c\in\mathcal{C}} d_c \tilde{y}_c^r \mu_r && s\in\mathcal{S} \\
&\quad u_S = \sum_{p\in P_S}\lambda_p && S\subseteq\mathcal{S}, S\neq\emptyset && (12)\\
&\sum_{s\in S} b_s \le Q_1 u_S && S\subseteq\mathcal{S}, S\neq\emptyset && (13)\\
&\quad \lambda_p\in\mathbb{N} && p\in P \\
&\quad \mu_r\in\{0,1\} && r\in R
\end{aligned}$$

Constraints (12) define variables u. Level balancing constraints (13) replace variables w and constraints (6), (7), (11). We now prove that F2 is a projection of F1. The proof is illustrated in Figure 1.

Proposition 1. *A solution $(\bar{\lambda},\bar{\mu})$ is feasible for the LP relaxation (LF2) of formulation F2 if and only if there exists a feasible solution $(\bar{\lambda},\bar{\mu},\bar{w})$ for the LP relaxation (LF1) of formulation F1.*

Proof. We first prove necessity, i.e "only if" part. Given a solution $(\bar{\lambda},\bar{\mu})$ for F2 and its computed values $(\bar{b},\bar{u})$, we construct a directed graph $\bar{G}=(\bar{V},\bar{A})$. Set $\bar{V}$ of nodes contains source $\tilde{s}$, sink $\tilde{t}$, set $\bar{P}=\{p\in P:\ \bar{\lambda}_p>0\}$ of nodes representing first-level routes in the solution, and set $\bar{S}=\{s\in\mathcal{S}:\ \bar{b}_s>0\}$ of nodes representing satellites used in the solution. Set $\bar{A}$ of arcs is the union of the following three sets: $\bar{A}_1$ connects the source with $\bar{P}$, $\bar{A}_2$ connects $\bar{P}$ with $\bar{S}$, and $\bar{A}_3$ connects $\bar{S}$ with the sink. An arc $(\tilde{s},p)$ in $\bar{A}_1$ has capacity $Q_1\bar{\lambda}_p$. An arc (p,s) belongs to $\bar{A}_2$ if and only if $s\in S_p$ and has infinite capacity. An arc $(s,\tilde{t})$ in $\bar{A}_3$ has capacity $\bar{b}_s$.

Let us now prove by contradiction that the maximum value of the $\tilde{s}$- $\tilde{t}$ flow in graph $\bar{G}$ is equal to $\sum_{c\in\mathcal{C}} d_c = d(\mathcal{C})$. Suppose that the maximum flow is strictly less than $d(\mathcal{C})$. Let $\bar{V}'$ be the subset of $\bar{V}$ obtained from a minimum $\tilde{s}$-$\tilde{t}$ cut in $\bar{G}$, $\tilde{s}\in\bar{V}'$. Let $\bar{S}'=\bar{S}\setminus\bar{V}'$ and $\bar{P}'=\bar{P}\setminus\bar{V}'$. We denote as $\delta(\bar{V}')=\{(v,v')\in\bar{A}\mid v\in\bar{V}', v'\notin\bar{V}'\}$ the set of arcs forming the minimum cut. From the supposition and the max-flow-min-cut theorem it follows that the total capacity of $\delta(\bar{V}')$ is less than $d(\mathcal{C})$. Thus $\delta(\bar{V}')$ contains at least one arc in $\bar{A}_1$ and does not contain all arcs in $\bar{A}_3$. Therefore, the total capacity of arcs in $\bar{A}_1\cap\delta(\bar{V}')$ is strictly less than the total capacity of arcs in $\bar{A}_3\setminus\delta(\bar{V}')$:

$$\sum_{p\in\bar{P}'} Q_1\bar{\lambda}_p < \sum_{s\in\bar{S}'}\bar{b}_s. \tag{14}$$

Set $\delta(\bar{V}')$ does not contain any arc in $\bar{A}_2$ as they have infinite capacity. Thus $\bar{V}'\cap\bar{P}\cap P_{\bar{S}'}$ is empty, and $\bar{\lambda}_p=0$ for all $p\in P_{\bar{S}'}\setminus\bar{P}'$. From the latter and (14), it follows that $\sum_{s\in\bar{S}'}\bar{b}_s > Q_1\sum_{p\in P_{\bar{S}'}}\bar{\lambda}_p$ which violates constraints (12) and (13) for set $\bar{S}'$ of satellites. Thus $(\bar{\lambda},\bar{\mu})$ is not a feasible solution for LF2 which leads to a contradiction.

We have just proved that the maximum flow in graph $\bar{G}$ has value $d(\mathcal{C})$. We now set $\bar{w}_{ps}$ equal to the flow from $p\in\bar{P}$ to $s\in\bar{S}$ for every $(p,s)\in\bar{A}_2$, and to 0 otherwise. By construction of graph $\bar{G}$, constraints (6) and (7) are satisfied by $(\bar{\lambda},\bar{b},\bar{w})$, and $(\bar{\lambda},\bar{\mu},\bar{w})$ is a feasible solution for LF1.

To show the sufficiency ("if" part), we prove that constraints (13) are valid for LF1. If $(\bar{\lambda},\bar{\mu},\bar{w})$ is feasible for LF1, then for an arbitrary subset of satellites $S\subseteq\mathcal{S}$, $S\neq\emptyset$, we have

$$\sum_{s\in S}\bar{b}_s \overset{(7)}{=} \sum_{s\in S}\sum_{p\in P_{\{s\}}}\bar{w}_{ps} = \sum_{p\in P_S}\sum_{s\in S_p}\bar{w}_{ps} \overset{(6)}{\le} \sum_{p\in P_S} Q_1\bar{\lambda}_p \overset{(12)}{\le} Q_1 u_S.$$

Thus, $(\bar{\lambda}, \bar{\mu})$ is feasible for LF2. $\square$

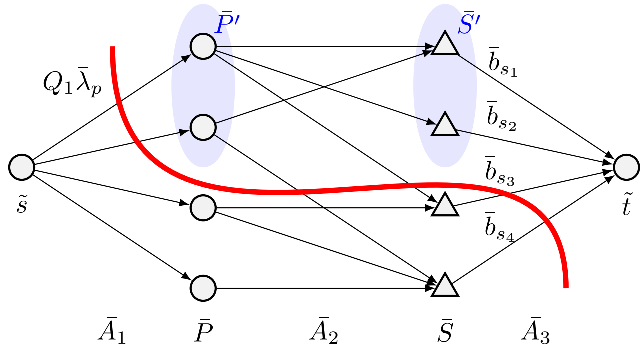

Figure 1: Illustration for graph $\bar{G}$ and its minimum cut in Proposition 1.

The main advantage of formulation (F2) is a significantly smaller number of variables. This comes at the cost of an additional exponential set of constraints (12) and (13). However, these constraints can be separated in polynomial time as showed in the proof of Proposition 1. In the separation procedure, we search for a minimum cut in graph $\bar{G}$ constructed from a fractional solution $(\bar{\lambda}, \bar{\mu})$ of (F2). Once set $\bar{S}'$ of satellites is found, it is further separated into subsets such that there is no path in $\bar{P}$ visiting two satellites in different subsets. Then, we add constraints (12) and (13) for every such subset of satellites.

An additional potential advantage of formulation (F2) is the possibility to generate variables λ_p dynamically, similarly to variables μ. Dynamic generation of λ is necessary when the number of satellites is large. We do not do it in our work, as we consider instances with at most 15 satellites. Instead, we enumerate all the first-level routes. In this procedure, for each subset $S \subseteq \mathcal{S}$ of satellites, we solve the traveling salesman problem in which we find the shortest route starting and finishing at the depot 0 and visiting all satellites in S exactly once. Then, we assign each first-level route to a λ variable, and we add all first-level route variables λ to the formulation before starting the algorithm. Thus, the formulation contains $2^{|\mathcal{S}|} - 1$ first-level route variables. All exact algorithms in the literature for the 2E-VRP use enumeration of first-level routes.

3.3 Valid inequalities

In our BCP algorithm, we use four families of valid inequalities. In this section, we present the first three families. To simplify the presentation, we introduce auxiliary variables x and y:

$$x_e^s = \sum_{r \in R_s} \tilde{x}_e^r \mu_r, \quad s \in \mathcal{S},\ e \in E_2, \qquad y_c^s = \sum_{r \in R_s} \tilde{y}_c^r \mu_r, \quad s \in \mathcal{S},\ c \in \mathcal{C}.$$

Integer variable x_e^s is equal to the number of times edge $e \in E_2$ is used by city freighters started from satellite $s \in \mathcal{S}$. Binary variable y_c^s is equal to one if and only if customer $c \in \mathcal{C}$ is visited by a city freighter started from satellite $s \in \mathcal{S}$.

3.3.1 Rounded capacity cuts

Rounded capacity cuts were introduced by Laporte and Nobert (1983) for the CVRP. Term $\lceil d(C)/Q_2 \rceil$ is a lower bound on the number of city freighters which have to visit at least one customer in C. Therefore, the next rounded capacity cuts (RCC) are valid:

$$\sum_{s \in \mathcal{S}} \sum_{e \in \delta_2(C)} x_e^s \geq 2 \left\lceil \frac{d(C)}{Q_2} \right\rceil, \quad C \subseteq \mathcal{C}. \tag{15}$$

Constraints (15) are separated using the CVRPSEP package (Lysgaard, 2018) which implements the heuristic by Lysgaard et al. (2004).

3.3.2 Chvátal-Gomory rank-1 cuts

Here we consider Chvátal-Gomory rounding of set-partitioning constraints (2) relaxed to $\leq$ inequalities. Consider a vector α of multipliers such that $\alpha_c \geq 0$, $c \in \mathcal{C}$. Then, the following rank-1 cut (R1C) is valid:

$$\sum_{s\in S}\sum_{r\in R_s}\left\lfloor \sum_{c\in\mathcal{C}} \alpha_c \tilde{y}_c^r \right\rfloor \mu_r \leq \left\lfloor \sum_{c\in\mathcal{C}} \alpha_c \right\rfloor . \tag{16}$$

An inequality (16) obtained using a vector of multipliers with l positive components is called an l-row rank-1 cut. If all positive components of α are the same, the corresponding inequality is called a subset-row cut. Jepsen et al. (2008) first introduced 3-row subset-row cuts. Pecin et al. (2017a) used l-row subset-row cuts with $l \leq 5$. General l-row rank-1 cuts with $l \leq 5$ were considered by Pecin et al. (2017b). They determined all dominant vectors of multipliers for such cuts: if an l-row rank-1 cut with $l \leq 5$ is violated, at least one rank-1 cut obtained using a dominant vector of multipliers is violated.

Similarly to Sadykov et al. (2017), separation of l-row rank-1 cuts with $l \leq 5$ is performed using a local search heuristic separately for every dominant vector of multipliers. We employ the limited memory technique by Pecin et al. (2017a) to reduce the impact of rank-1 cuts on the solution time of the pricing problem.

3.3.3 Visited satellite inequalities

A customer can be visited by a route $r \in R_s$ only if satellite s is visited by at least one urban truck. Therefore, next visited satellite inequalities (VCI) are valid:

$$y_c^s \leq u_{\{s\}}, \quad c \in \mathcal{C}, s \in \mathcal{S}. \tag{17}$$

Although inequalities (17) are rather straightforward, we did not find any work in the literature which uses them. Separation of constraints (17) is performed by enumeration of all pairs $(c, s) \in \mathcal{C} \times \mathcal{S}$.

3.3.4 Lower bounds on the number of vehicles

We define lower bounds on the number of urban trucks

$$\sum_{p\in P} \lambda_p \geq \left\lceil \frac{d(\mathcal{C})}{Q_1} \right\rceil \tag{18}$$

and the number of city freighters

$$\sum_{r\in R} \mu_r \geq \left\lceil \frac{d(\mathcal{C})}{Q_2} \right\rceil \tag{19}$$

Moreover, a subset S of satellites must be visited by enough urban trucks to supply the demand that cannot be delivered from satellites in $\mathcal{S} \setminus S$. Therefore, the next lower bound on u_S is valid :

$$u_S \geq \left\lceil \frac{d(\mathcal{C}) - \sum_{s\in\mathcal{S}\setminus S} L_s Q_2}{Q_1} \right\rceil, \quad S \subset \mathcal{S} \tag{20}$$

Inequalities (20) are useful when the number of city freighters that can start from satellites is limited ($L_s < |\mathcal{L}|, s \in \mathcal{S}$).

4 New family of valid inequalities

We propose a new family of satellite supply inequalities (SSI) inspired by the depot capacity constraints introduced for the capacitated location-routing problem by Belenguer et al. (2011). To simplify the presentation below, we denote $C^{\complement} = \mathcal{C} \setminus C$ and $S^{\complement} = \mathcal{S} \setminus S$. Let us introduce SSI using an example.

Example 1. *Consider urban trucks with capacity $Q_1 = 10$ and city freighters with capacity $Q_2 = 6$. Figure 2 shows a fractional solution $(\bar{u}, \bar{\mu})$ for LF2. Here $\mathcal{S} = \{s_1, s_2\}$ and set $\mathcal{C}$ contains seven customers. Consider subset C of customers with $d(C) = 11$. Consider also subset $S = \{s_2\}$ of satellites with $\bar{u}_S = 1$. Clearly, S can supply only demand of at most $Q_1\bar{u}_S = 10$ units and thus cannot supply set C of customers alone. In this fractional solution, C is supplied by 1.8 city freighters coming from S and 0.2 city freighters coming from $S^{\complement}$. The violated SSI states that either two or more urban trucks should visit $S = \{s_2\}$ or at least one city freighter coming from satellites in $S^{\complement} = \{s_1\}$ should visit some customers in C:*

$$\sum_{s \in S^{\complement}} \sum_{e \in \delta(C)} x_e^s \geq 2 \cdot (2 - u_S)\,.$$

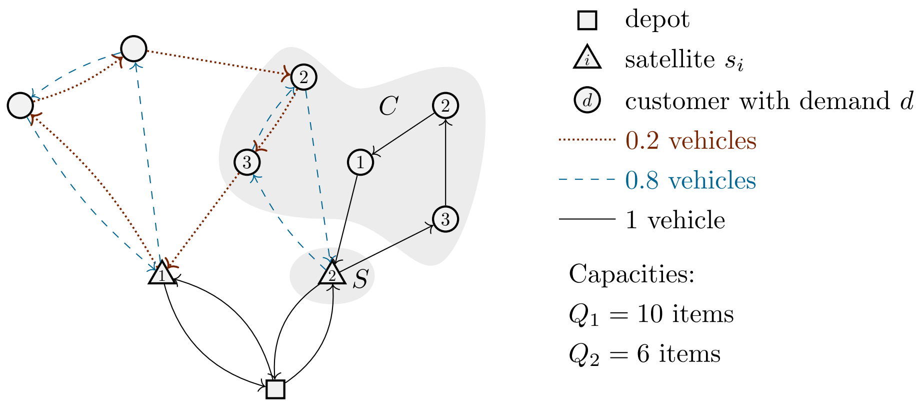

Figure 2: Example of a satellite supply inequality, violated for given S and C.

4.1 Satellite supply inequalities

We now define $g^C(u)$ as the function which gives a lower bound on the number of city freighters required to cover the demand of a subset C of customers that $\lfloor u \rfloor$ urban trucks cannot supply:

$$g^C(u) = \max\left\{0, \left\lceil \frac{d(C) - Q_1\lfloor u \rfloor}{Q_2} \right\rceil \right\}.$$

Proposition 2. *Given $C \subset \mathcal{C}$ and $S \subset \mathcal{S}$, the following inequality is valid for the 2E-CVRP*

$$\sum_{s \in S^{\complement}} \sum_{e \in \delta_2(C)} x_e^s \geq 2 \cdot g^C(u_S). \tag{21}$$

Proof. Consider a feasible solution $(\bar{x}, \bar{b}, \bar{u})$ of formulation (F2). The following rounded capacity inequality is satisfied by $\bar{x}$ and $\bar{b}$:

$$\sum_{s \in S^{\complement}} \sum_{e \in \delta_2(C)} \bar{x}_e^s \geq 2 \left\lceil \frac{d(C) - \sum_{s \in S} \bar{b}_s}{Q_2} \right\rceil \tag{22}$$

From constraint (13) and the integrality of variable $\bar{u}_S$, it follows

$$\sum_{s \in S} \bar{b}_s \leq Q_1 \bar{u}_S = Q_1 \lfloor \bar{u}_S \rfloor. \tag{23}$$

By combining (22) and (23) we obtain that (21) is satisfied by $\bar{x}$ and $\bar{u}$. $\square$

Function $g^C(u)$ is not linear and cannot be used directly. Instead, we use the piecewise linear function, denoted as $h^C(u)$, which forms the convex hull of the epigraph of $g^C(u)$. We denote as $\tilde{u}^C$ the ordered vector of (integer) values u of extreme points of h^C:

$$\tilde{u}^C = \left(\tilde{u}_0^C = 0, \tilde{u}_1^C, \ldots, \tilde{u}_{k(C)}^C = \lceil d(C)/Q_1 \rceil \right).$$

Figure 3 depicts an example of functions g^C and h^C for $Q_1 = 10$, $Q_2 = 4$, and $d(C) = 32$. In the left plot, the epigraph of g^C is the grey area. In the right plot, function h^C is the bold line. Extreme points of h^C are H_0, H_2, H_3, H_4, but not H_1. Therefore, $\tilde{u}^C = (0, 2, 3, 4)$, and $k(C) = 3$.

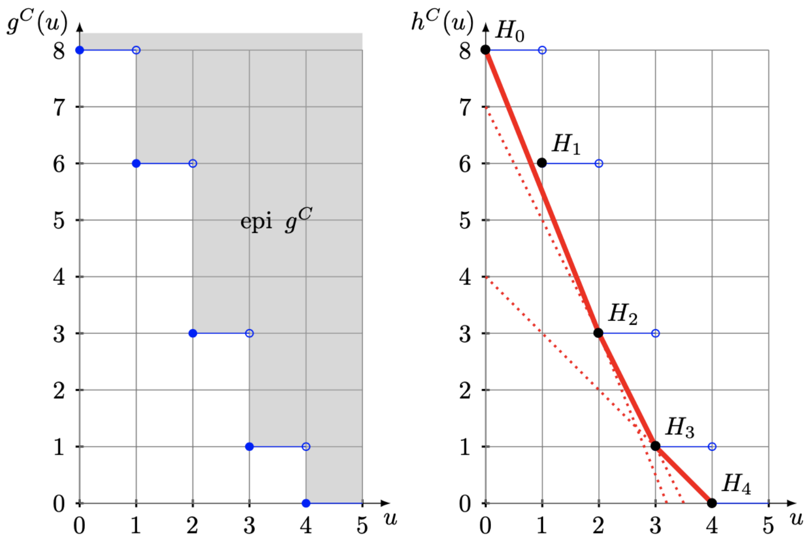

Figure 3: Example of functions g^C (on the left) and h^C (on the right).

Proposition 3. *Given subsets $C \subset \mathcal{C}$, $S \subset \mathcal{S}$, and an integer $0 < k \leq k(C)$, the following inequality is valid for the 2E-CVRP*

$$\sum_{s \in S^\complement} \sum_{e \in \delta_2(C)} x_e^s \geq 2 \cdot \left(h^C(\tilde{u}_{k-1}^C) - \frac{h^C(\tilde{u}_{k-1}^C) - h^C(\tilde{u}_k^C)}{\tilde{u}_k^C - \tilde{u}_{k-1}^C} (u_S - \tilde{u}_{k-1}^C) \right) \tag{24}$$

Proof. The right-hand side of each constraint (24) corresponds to a linear piece of function h^C. Thus the proof follows from Proposition 2 and from the fact that $h^C(u) \leq g^C(u)$ for all $u \geq 0$. $\square$

4.2 Separation of SSI

Let $(\bar{x}, \bar{u})$ be the values of variables x and u in a solution for LF2. The following problem finds the most violated inequality.

$$\max_{S \subset \mathcal{S}, C \subset \mathcal{C}} 2 \cdot h^C(\bar{u}_S) - \sum_{s \in S^\complement} \sum_{e \in \delta_2(C)} \bar{x}_e^s \tag{25}$$

The first and second terms of the objective function are non-linear functions of C and S. Thus enumeration of C and S is required to compute (25) exactly using an integer program. Since the number of subsets is exponential, we propose a heuristic to separate SSI. Although the heuristic does not necessarily find the most violated inequality, it offers a good trade-off between the computational effort and the violation of found inequalities. Our heuristic works with a fixed set of satellites. The following proposition gives a dominance rule which will allow us to discard non-interesting subsets of satellites.

Proposition 4. *Consider a solution $(\bar{x}, \bar{u})$ for LF2 and a fixed set C of customers. If SSI (24) is violated for C and a set S_1 of satellites, then it is violated for C and any set $S_2 \supseteq S_1$ such that $\bar{u}_{S_2} = \bar{u}_{S_1}$.*

Proof. The right-hand side of (24) is fixed for a fixed set C of customers and a fixed value u_S. Thus right-hand side of the SSIs for pairs (C, S_1) and (C, S_2) is the same. For fixed values of variables x, the left-hand side of the SSI for pair (C, S_2) is not larger than one of the SSI for pair (C, S_1), as $S_2^{\complement} \subseteq S_1^{\complement}$. Thus the violation of the SSI for pair (C, S_2) is not smaller than one of the SSI for pair (C, S_1). $\square$

To enumerate all the non-dominated sets of satellites, we first build the power set $\bar{\mathcal{U}}$ of set $\bar{S}$ of satellites used in the solution. Since we look for the largest subsets of satellites, we append all satellites in $\mathcal{S} \setminus \bar{S}$ to each set in $\bar{\mathcal{U}}$. Finally, we exclude from $\bar{\mathcal{U}}$ all sets S_1 such that there exists $S_2 \in \bar{\mathcal{U}}$ with $S_2 \supseteq S_1$ and $\bar{u}_{S_2} = \bar{u}_{S_1}$. This is done by the exhaustive enumeration as the cardinality of set $\bar{\mathcal{U}}$ is not large for the instances of the literature.

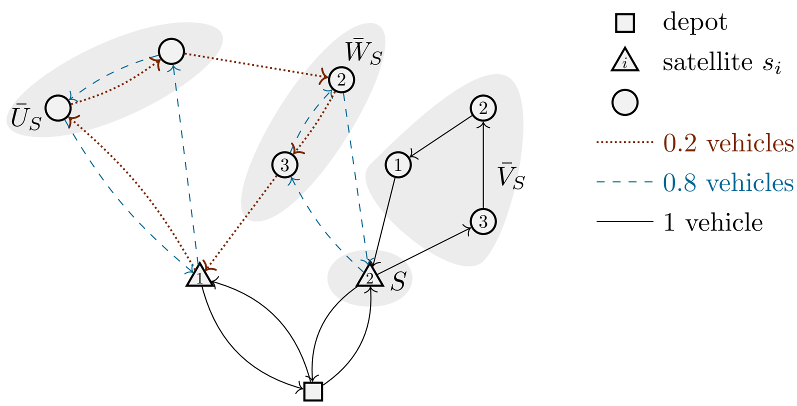

Figure 4: Separation graph $\bar{G}_2(S)$ for the fractional solution in Example 1

Given a set $S \in \bar{\mathcal{U}}$ of satellites, we now look for subsets of customers that violate SSI. We split customers in three subsets. Let $\bar{U}_S \subset \mathcal{C}$ be the set of customers visited only by routes started from $S^{\complement}$, let $\bar{V}_S \subset \mathcal{C}$ be the set of customers visited only by routes started from S, and let $\bar{W}_S = \mathcal{C} \setminus (\bar{U}_S \cup \bar{V}_S)$. Figure 4 illustrates the partition of customers of Example 1. We then build graph $\bar{G}_2(S)$ in the following way.

- Graph $\bar{G}_2(S)$ is the subgraph of G_2 induced by vertices in $\bar{U}_S \cup \bar{W}_S \cup S^{\complement}$.
- Weight of each edge e in $\bar{G}_2(S)$ is equal to $\sum_{s \in S^{\complement}} \bar{x}_e^s$.
- Set $\bar{U}_S \cup S^{\complement}$ of vertices in $\bar{G}_2(S)$ is merged into one vertex $\bar{s}$ by successively contracting all edges having two incident vertices in $\bar{U}_S \cup S^{\complement}$. Weight of each edge $(\bar{s}, c)$, $c \in \bar{W}_S$, in final graph $\bar{G}_2(S)$ is then equal to $\sum_{s \in S^{\complement}} \sum_{i \in \bar{U}_S \cup S^{\complement}} \bar{x}_{(i,c)}^s$.

Afterwards, we compute the minimum cut in $\bar{G}_2(S)$. Let $\bar{C}_S$ be the subset of vertices obtained from the minimum cut, $\bar{s} \notin \bar{C}_S$. First, we verify whether SSI based on set S of satellites and set $C = \bar{C}_S \cup \bar{V}_S$ of customers is violated. Then, we iteratively enlarge set C in a greedy manner and check the violation of SSI based on S and C at each iteration. Customer c' to include in current set C (and exclude from $\bar{W}_S$) in each iteration is

$$c' = \arg\max_{c \in \bar{W}_S} \left\{ \frac{\sum_{s \in S^{\complement}} d_c \bar{y}_c^s}{\sum_{s \in S^{\complement}} \sum_{i \in \bar{U}(S,C,c)} \bar{x}_{(c,i)}^s} \right\}, \tag{26}$$

where $\bar{U}(S,C,c) = S^{\complement} \cup \bar{U}_S \cup \bar{W}_S \setminus \{c\}$. The intuition behind (26) is that we try to increase the first term of (25) while increasing not too much the second term. The separation procedure is formally given in Algorithm 1.

Algorithm 1 Separation procedure for the satellites supply inequalities

We are given $(\bar{\lambda}, \bar{\mu})$ and computed entities $\bar{P}$, $\bar{u}$, $\bar{x}$, $\bar{y}$
$\mathcal{I}$ is the set of found violated SSI
Find set $\bar{\mathcal{U}}$ of non-dominated subsets of satellites
for all $S \in \bar{\mathcal{U}}$ **do**
  Find $\bar{U}_S$, $\bar{V}_S$, $\bar{W}_S$ and build graph $\bar{G}_2(S)$
  Compute the minimum cut in $\bar{G}_2(S)$ and corresponding set $\bar{C}_S$
  $C \leftarrow \bar{C}_S \cup \bar{V}_S$
  repeat
    If SSI based on S and C is violated by $(\bar{u}, \bar{x})$, add the inequality to $\mathcal{I}$
    Find customer c' in $\bar{W}_S$ using (26)
    $C \leftarrow C \cup \{c'\}$
    $\bar{W}_S \leftarrow \bar{W}_S \setminus \{c'\}$
  until $\bar{W}_S = \emptyset$
end for
Return a specified number of the most violated SSI in $\mathcal{I}$

We now analyse the complexity of Algorithm 1. The cardinality of set $\bar{\mathcal{U}}$ is at most $2^{|\bar{S}|}$, as well as the number of iterations in the separation algorithm. Construction of graph $\bar{G}_2(S)$ takes at most $|\bar{S}| \cdot |\mathcal{C}|^2$ operations. Computing the minimum cut has complexity $O(|\mathcal{C}|^3)$. The number of iterations of the internal loop is at most $|\mathcal{C}|$, and the complexity of each iteration is $|\bar{S}| \cdot |\mathcal{C}|^2$. Thus, the overall complexity of the separation algorithm is $O\big(2^{|\bar{S}|} \cdot |\bar{S}| \cdot |\mathcal{C}|^3\big)$.

5 Branch-Cut-and-Price algorithm

Formulation F2 together with valid inequalities RCC, R1C, VCI, and SSI is solved by an adaptation of the branch-cut-and-price algorithm proposed by Sadykov et al. (2017). In this section, we describe the main ingredients of this algorithm. The reader is invited to consult the original paper for all details.

As the number of variables depends exponentially on the number of satellites and customers, LF2 (called the *master problem*) is solved by the column and cut generation approach.

5.1 Pricing problem

Second-level route variables μ are dynamically generated by solving the pricing problem. It is decomposed into $|\mathcal{S}|$ subproblems (SP$_s$), one per satellite $s \in \mathcal{S}$. The set of feasible solutions for SP$_s$ is the set R_s of paths in graph G_2. A path r belongs to R_s if and only if :

- it starts and finishes in vertex s: $\sum_{e \in \delta_2(s)} \tilde{x}_e^r = 2$;
- it does not pass through other satellites: $\sum_{e \in \delta_2(\mathcal{S} \setminus \{s\})} \tilde{x}_e^r = 0$;
- it passes through each customer at most once: $\tilde{y}_c^r \leq 1$, $\forall c \in \mathcal{C}$.

- its total delivered demand does not exceed the capacity of a city freighter: $\sum_{c\in\mathcal{C}} \tilde{y}^r_c \leq Q_2$.

Let π, ψ, ϕ, ρ, τ, θ, ξ, and ζ be optimum dual values for the master problem restricted to a subset of variables μ. These dual values correspond to constraints (2), (3), (4), (8), (15), (16), (17), and (24). Let also K be the collection of active rank-1 cuts corresponding to vectors $(\alpha^k)_{k\in K}$, and M_s be the collection of active SSI based on sets $(S_m, C_m)_{m\in M_s}$, $s \in S^{\complement}_m$. The reduced cost of a path $r \in R_s$ is then equal to

$$\sum_{e\in E_2}\left(f^T_e - \sum_{\substack{C\in\mathcal{C}:\\ e\in\delta(C)}} \tau_C - \sum_{\substack{m\in M_s:\\ e\in\delta_2(C_m)}} \zeta_m\right)\tilde{x}^r_e$$
$$-\sum_{c\in\mathcal{C}}\sum_{e\in\delta_2(c)} \frac{1}{2}\left(\pi_c + d_c\rho_s - \xi^s_c\right)\tilde{x}^r_e + \sum_{k\in K}\theta_k\left\lfloor\sum_{c\in\mathcal{C}}\alpha^k_c\tilde{y}^r_c\right\rfloor + \psi_s + \phi. \tag{27}$$

When rank-1 cuts (16) are absent, a path has a reduced cost that is equal to the sum of reduced costs of its edges. In this case, pricing subproblem SP_s is the resource constrained elementary shortest path problem. This problem can be time-consuming to solve. Therefore, we relax the elementarity constraint and work with ng-paths (Baldacci et al., 2011). This relaxation has an impact only on the optimum value of LF2. The validity of F2 is preserved. For each customer, its ng-neighborhood is static and includes 8 closest customers including itself.

Presence of active rank-1 cuts makes the pricing problem more complicated because each such cut adds a resource to the problem. To limit the increase in difficulty of the pricing problem, we use the limited memory technique proposed by Pecin et al. (2017a). For each cut (16) a memory is associated. It consists of a subset of edges in E_2. This memory makes the resource associated to the cut local: we need to track its consumption only in a small part of the graph.

Each pricing subproblem SP_s is solved by the bucket graph based bi-directional labeling algorithm proposed by Sadykov et al. (2017). This algorithm supports both ng-path relaxation and presence of dual values associated with limited-memory rank-1 cuts. If the transportation costs f^T are symmetric, the algorithm exploits the forward-backward path symmetry.

5.2 Column and cut generation

We use three-stage column generation to speed up convergence. In the first two stages, the pricing problem is solved by the same heuristic labelling algorithms as in (Sadykov et al., 2017). In the last stage, the pricing problem is solved by the exact labelling algorithm. For each subproblem, heuristic pricing generates at most 30 columns, and exact pricing generates at most 150 columns. We also apply automatic dual pricing smoothing stabilization suggested by Pessoa et al. (2018) to further speed-up the convergence of column generation.

We use the bucket arc elimination procedure (Sadykov et al., 2017). It reduces the size of graph G_2 by removing arcs which are proved to be absent from any improving solution to the problem. The procedure is first performed after the first column generation convergence, and then each time the primal-dual gap decreases by more than 10%. Note that graph reduction is not the same in different pricing subproblems.

After each call to the bucket arc elimination procedure, we use the elementary route enumeration technique by Baldacci et al. (2008). For each pricing subproblem, this procedure tries to enumerate all elementary routes with reduced cost smaller than the current primal-dual gap. If the enumeration succeeds, the pricing subproblem is solved by inspection in future column generation iterations, similarly to Contardo and Martinelli (2014). If the total number of enumerated routes in all subproblems is less than 5000, all the routes are added to formulation F2 and the latter is solved by the MIP solver.

We define variables u only for subsets with 5 satellites or less in order to limit the size of the formulation and reduce the number of candidates for branching. Moreover, for every subset

S such that $|S| \geq 6$, we replace variable u_S by expression $t_1 - \sum_{p \notin P_S} \lambda_p$, where t_1 is the total number of urban trucks used. This replacement allows us to keep the coefficient matrix sparse. In the beginning, all constraints (13) are added to formulation F2 for instances with at most 10 satellites. For instances with 15 satellites, only constraints (13) for sets $S \subset \mathcal{S}$, $|S| \leq 5$, are added. Other constraints are dynamically separated as described in Section 3.2.

In each cut generation round, we add at most 100 rounded capacity cuts (15), 450 rank-1 cuts (16), 50 VCI (17), and 150 SSI (24) to the master problem. The cut generation is stopped either by the tailing-off condition or when the time spent to solve at least one pricing subproblem exceeds 1 second. The tailing-off condition is satisfied when after 3 cut generation rounds the primal-dual gap decreases by less than 2% per round.

The parameterisation of column and cut generation is inherited from the BCP algorithm in Sadykov et al. (2017).

5.3 Branching

We perform branching on: the number of first-level routes visiting a subset of satellites (variables u), the use of first-level routes (variables λ), the use of an edge in E_2 (variables x), the assignment of a customer to a satellite (variable y), the number of first-level routes ($\sum_{p \in P} \lambda_p$), the number of second-level routes ($\sum_{r \in R} \mu_r$), and the number of second-level routes started from a satellite s ($\sum_{r \in R_s} \mu_r$). We use a multi-phase strong branching procedure, similar to Sadykov et al. (2017), to choose the most promising branching candidate.

The branching procedure first chooses at most 50 branching candidates. Up to half of the candidates are chosen according to the branching history using pseudo-costs. In the first phase, we solve the LP relaxation of the restricted master problem. Three candidates with the largest product of lower bound improvements in the branches are chosen for the next phase. In the second phase, column generation is performed, but the pricing problem is solved with only heuristic labeling algorithms. The best candidate is chosen using the same product rule. In the third phase, the exact column and cut generation is performed in both branches of the chosen candidate using the same parameters as in the root node.

5.4 Primal heuristic

After each node in the branch-and-bound tree, a heuristic looks for improving feasible solutions to the 2E-CVRP. We first tried the standard restricted master heuristic (Sadykov et al., 2018) in which the MIP solver solves the current restricted master problem. However, we have not been satisfied with the performance of this heuristic, especially for instances with large capacity of city freighters. The reason is that sometimes only a small part of columns in the restricted master are elementary, thus making the solution space of the restricted master very small or even empty.

Instead, we use the heuristic based on an artificial primal bound and the elementary route enumeration, similar to Pessoa et al. (2009). In an iterative procedure, we decrease the artificial bound in order to divide the primal-dual gap by two in each iteration. Then, we perform elementary route enumeration for each pricing subproblem. The iterative procedure stops when the enumeration succeeds for all subproblems. Afterward, we pick 10000 elementary routes with the smallest reduced cost and add them to the master problem. Finally, we let IBM CPLEX MIP solver to solve the resulting problem with the time limit of $|\mathcal{C}|/2.5$ seconds. We activate the polishing heuristic (Rothberg, 2007) implemented in CPLEX.

6 Computational Results

The implementation of proposed algorithm is done using VRPSolver by Pessoa et al. (2019), which is available at `https://vrpsolver.math.u-bordeaux.fr` for free academic use. This solver includes the following components:

- BaPCod C++ library (Vanderbeck et al., 2019) which implements the BCP framework;
- C++ code, developed by Sadykov et al. (2017), which implements the bucket graph based labeling algorithm, bucket arc elimination procedure, elementary route enumeration, and the separation algorithm for R1C;
- CVRPSEP C++ library (Lysgaard, 2018) which implements heuristic separation of RCC.

IBM CPLEX Optimizer version 12.8.0 is used as the LP solver in column generation and as the solver for the enumerated MIPs.

To reproduce the results, it suffices to i) define models F1 and F2; ii) to implement separation algorithms for the exponential set of constraints (12) and (13), and valid inequalities VCI and SSI. For other components, VRPSolver and CPLEX can be used. Our implementation of models and separation algorithms is done in Julia 0.6 language using JuMP (Dunning et al., 2017), LightGraphs and VRPSolver packages.

Experiments were run on a 2 Dodeca-core Haswell Intel Xeon E5-2680 v3 servers at 2.5 GHz. On each server, we solved 24 instances with up to 200 customers and up to 10 satellites that share 128Go of RAM. Larger instances having either 300 customers or 15 satellites were solved by batches of 4 instances sharing 128Go of RAM. Each instance is solved on a single thread.

6.1 Instances

Table 1 shows the sets of instances from the literature that we used. Constraints (3) limiting the number of city freighters per satellite are required only for set 4B. Therefore, constraints (20) are useful only for set 4B. Set 5 duplicates each instance: the first instance has the standard capacity of city freighters, and the second one, with suffix “b”, has the double capacity. Only set 6B has non-zero handling costs. We do not consider instances of set 3, proposed in (Gonzalez-Feliu et al., 2007), as they are easily solved by our algorithm and by Baldacci et al. (2013).

Table 1: Sets of instances from the literature used for experiments

Set	#	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{C}\rvert$	Notes	Authors
4A	54	2, 3, 5	50	$L_s < \lvert\mathcal{L}\rvert$	Crainic et al. (2010)
4B	54	2, 3, 5	50		Crainic et al. (2010)
5	18	5, 10	100, 200	low and high Q_2	Hemmelmayr et al. (2012)
6A	27	4, 5, 6	50, 75, 100		Baldacci et al. (2013)
6B	27	4, 5, 6	50, 75, 100	$f_s^H > 0$	Baldacci et al. (2013)

We also generated 51 additional instances involving up to 300 customers and 15 satellites. They are based on instances of families a, b, and c proposed by Schneider and Löffler (2019) for the capacitated location-routing problem. In comparison with the original instances, we added the position of the depot, capacity of urban trucks, the number of urban trucks, and the number of city freighters. We put the depot at location $(0, 0)$. We set $Q_1 = 9 \cdot Q_2$, $\lvert\mathcal{K}\rvert = \lceil 1.75 \cdot d(\mathcal{C})/Q_1 \rceil$, and $\lvert\mathcal{L}\rvert = \lceil 2.5 \cdot d(\mathcal{C})/Q_2 \rceil$. New set of instances available at `www.math.u-bordeaux.fr/~rsadykov/tests/2E-VRP-set7.zip`.

6.2 Experimental analysis of BCP variants

In the first experiment, we compare different variants of our BCP algorithm for solving the 54 largest literature instances from sets 5, 6A, and 6B with 75, 100, and 200 customers. To eliminate randomness related to improvement of primal bounds, all variants were executed without primal heuristic and with the initial primal bound equal to the optimum solution value (plus small ϵ) for each instance. We tested the following BCP variants.

F1+RCC+R1C — the base variant which does not separate new valid inequalities VCI and SSI and does not branch on u. This variant can be considered as a straightforward adaptation of the BCP algorithm in (Sadykov et al., 2017) for solving the 2E-CVRP (i.e. the direct application of VRPSolver to the 2E-CVRP). This variant borrows all recent advances for classical problems like CVRP.

F1+RCC+R1C+u — the base variant with branching on variables u.

F2+RCC+R1C+VCI+SSI+u — the best variant, which is based on formulation (F2), with separation of VCI and SSI, and with branching on variables u.

F2+RCC+R1C+VCI+SSI — the best variant without branching on variables u.

F2+RCC+R1C+SSI+u — the best variant without separating VCI.

F2+RCC+R1C+VCI+u — the best variant without separating SSI.

F1+RCC+R1C+VCI+SSI+u — the best variant, but based on formulation (F1).

Table 2 gives the comparison of BCP variants. It contains average values for the root gap, geometric mean values for the root solution time, the number of branch-and-bound nodes, the geometric mean of the master problem solution time, the geometric mean of total solution time in seconds, and the number of instances solved within the time limit set to 3 hours. For unsolved instances, the solution time is set to the time limit.

Table 2: Comparison of variants of the BCP algorithm

Variant	Root Gap (%)	Root Time (s)	Nodes	Master Time (s)	Total Time (s)	Solved
F1+RCC+R1C	4.29	83.6	76.0	713.7	2333.0	31/54
F1+RCC+R1C+u	4.28	99.5	24.5	338.9	1020.5	44/54
F1+RCC+R1C+VCI+SSI+u	0.68	177.8	5.9	96.2	421.4	49/54
F2+RCC+R1C+VCI+u	1.64	115.5	12.5	88.4	426.1	49/54
F2+RCC+R1C+SSI+u	0.71	238.6	6.6	85.8	501.0	50/54
F2+RCC+R1C+VCI+SSI	0.67	161.2	7.0	63.0	384.7	50/54
F2+RCC+R1C+VCI+SSI+u	0.68	159.0	6.3	59.0	361.7	51/54

We see that the base variant F1+RCC+R1C is the worst as it solves only 31 out of 54 instances. Adding branching on variables u improves significantly the base variant as variant F1+RCC+R1C+u solves 13 more instances. The best variant F2+RCC+R1C+VCI+SSI+u solves 7 more instances to optimality within the time limit. Table 2 shows that all our contributions improve the efficiency of the BCP algorithm. The root gap decreases significantly when new valid inequalities VCI and SSI are separated. Although the root solution time increases when additional inequalities are separated, the overall time decreases due to the much smaller size of the branch-and-bound tree. Thus, branching on variables u has a small effect on the performance of variant F2+RCC+R1C+VCI+SSI+u. However, adding branching on variables u is the simplest way to make the base variant F1+RCC+R1C much more efficient.

If one uses formulation F2 instead of F1, i) the cumulative time to solve the restricted master LP reduces, as the number of variables decreases; ii) the root gap however does not change, as it is expected from Proposition 1 which shows equivalence of LF1 and LF2; iii) two more instances are solved to optimality; iv) the average reduction of the solution time is relatively small, as the cumulative time to solve the restricted master LP is minor in comparison with the total solution time.

6.3 Comparison with the state-of-the-art algorithm

Let us now compare the variants F1+RCC+R1C and F2+RCC+R1C+VCI+SSI+u to the best exact algorithm by Baldacci et al. (2013). For a fair comparison, we do not use initial primal bounds in this experiment. Instead, we rely on the primal heuristic presented in section 5.4 to find feasible solutions. Baldacci et al. (2013) did not set an overall time limit for their algorithm. They set a limit on the number of collections of first-level routes considered, as well as a time limit of 5000 seconds for solving each subproblem with fixed subset of first-level routes. In our BCP algorithm, we set the time limit to 10 hours.

Table 3 shows the summary results of this experiment. For each set of literature instances, we give the average gap between the root dual bound and the best primal bound found (Rg), the geometric mean of the number of branch-and-bound nodes (Nds), the geometric mean of the solution time in seconds (t), and the number of instances solved to optimality (Solved). For a fair comparison, the solution time of (Baldacci et al., 2013) is divided by 1.6 because of the difference in computer speeds. The algorithm in (Baldacci et al., 2013) was tested only on 6 out of 18 instances of set 5. It has not been applied for instances with 10 satellites, as it is based on an enumeration of subsets of first-level routes. Since our algorithms are free from this drawback, they were tested on all instances of set 5.

Table 3: Comparison of two BCP variants with the state-of-the-art exact algorithm for the 2E-CVRP Baldacci et al. (2013)

	F1+RCC+R1C				F2+RCC+R1C+VCI+SSI+u				Literature	
Set	Rg(%)	Nds	t (s)	Solved	Rg(%)	Nds	t (s)	Solved	t (s)	Solved
4A	5.76	14.2	772	51/54	0.91	3.3	144	54/54	271	50/54
4B	4.45	12.7	550	52/54	0.98	3.6	203	54/54	232	52/54
5	5.83	222.9	20612	6/18	1.41	22.5	3215	15/18	8405	3/6
6A	7.04	99.7	2604	24/27	0.89	4.9	233	27/27	802	22/27
6B	3.15	57.8	1562	24/27	0.46	4.3	196	27/27	513	19/27

The base variant F1+RCC+R1C solves to optimality more instances than the best algorithm in the literature. However, the running time of the latter is on average smaller. The variant F2+RCC+R1C+VCI+SSI+u largely outperforms both other algorithms for all sets of instances. Indeed, our best BCP solves to optimality 31 open instances within 10 hours. Detailed results for this variant are given in A. The remaining 3 open instances were solved to optimality by providing the best-known solution of the literature as initial primal bound and using a special parameterisation. Detailed results for these instances are given in B.

Table 4 shows that our BCP algorithm could improve 10 best-known solutions (BKS) for literature instances. Their optimum solution values are given in column Opt. The improvement (Imp) is generally small. Thus, the existing heuristics for the 2E-CVRP have very good quality (at least when applied to literature instances).

6.4 Experimental results for new instances

As all instances from the literature were solved to optimality, we generated a new set of larger instances. The main goal was to test the scalability of our best algorithm, i.e. to determine the size of instances which cannot be solved in a reasonable time.

The new instances involve 5 – 15 satellites and 100 – 300 customers. When testing our algorithm on these instances, we set the time limit to 60 hours. We gave more time to the

Instance		BKS	Reference	Opt	Imp (%)
Set 5	100-5-1b	1103.55	Amarouche et al. (2018)	1099.35	0.38
	100-10-3b	849.73	Amarouche et al. (2018)	848.16	0.19
	200-10-1	1538.35	Amarouche et al. (2018)	1537.52	0.05
	200-10-1b	1175.81	Amarouche et al. (2018)	1173.07	0.23
	200-10-3	1779.68	Amarouche et al. (2018)	1177.49	0.12
	200-10-3b	1196.93	Amarouche et al. (2018)	1192.35	0.38
Set 6A	C-n101-4	1297.42	Wang et al. (2017)	1292.04	0.41
Set 6B	B-n101-4	1500.55	Breunig et al. (2016)	1499.71	0.06
	B-n101-5	1395.32	Breunig et al. (2016)	1394.79	0.04
	C-n101-5	1964.63	Breunig et al. (2016)	1962.52	0.11

Table 4: Improved best-known solutions for the literature instances

primal heuristic when solving the largest instances. For instances with 300 customers and 10 satellites, this time was set to 600 seconds. For instances with 15 satellites, this time was set to $4 \cdot |\mathcal{C}|$ seconds.

Out of 51 instance, our algorithm solved to optimality 23 instances, including some instances with 300 customers or with 15 satellites. The algorithm found both dual and primal bounds for 17 instances. The primal heuristic did not find any feasible solution for 9 instances having 300 customers and/or 15 satellites. Only lower bounds are thus currently known for these instances. We could not obtain dual bounds for 2 instances because the LP solver spent more than one hour to solve the restricted master LP during the first column generation convergence. Detailed results are given in C.

From the results on new instances, we conclude that our algorithm can solve to optimality instances up to 200 customers and 10 satellites, although sometimes in a long run. Our results on new instances up to 200 customers and 10 satellites are consistent with those on literature instances, even if one of these new instances was not solved to optimality. Increasing the instance size to either 300 customers or to 15 satellites makes our algorithm much less efficient. The instances with 300 customers and 15 satellites are particularly challenging, as none of them were solved to optimality.

For the largest instances with 15 satellites, the size of the restricted master problem becomes very large so that its solution takes significant time. In some extreme cases, a modern LP solver cannot solve it within 1 hour. Even when the LP relaxation of the restricted master for large instances can be solved, the primal heuristic based on solving the restricted master is often inefficient. One of the possible remedies is to generate first-level routes dynamically.

7 Conclusions

In this paper we proposed an improved branch-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. Our BCP algorithm includes both techniques for the classic vehicle routing problems proposed recently and new problem-specific components such as new route based formulation, two families of valid inequalities, and a new branching strategy. The proposed algorithm is empirically shown to be highly efficient, as it solved all instances available in the literature for the 2E-CVRP with up to 200 customers and 10 satellites. 34 instances were solved to optimality for the first time.

In order to inspire further progress on solution approaches for the 2E-CVRP and related problems, a new set of 51 instances is proposed to the community. Among them, 28 instances are currently open. Testing our algorithm on new instances revealed its limitations.

Dynamic generation of first-level routes is essential if one wants to solve instances with more than 15 satellites. Our new formulation for the problem without product flow variables allows one to do such dynamic generation. However, column generation of first-level routes is not straightforward for the modified formulation. A first-level route has coefficient one in constraints (12) if and only if it visits at least one satellite in a certain set. Thus, these constraints resemble strong capacity constraints introduced in (Baldacci et al., 2008). They are non-robust (Pessoa et al., 2008), i.e. they modify the structure of the pricing problem for the first level.

Our BCP algorithm could be extended to the two-echelon location-routing problem (Contardo et al., 2012) in which satellites have predefined capacities and fixed opening costs. Our preliminary research showed that additional valid inequalities are necessary for this variant of the problem. If one separates only the valid inequalities considered in this paper, the LP relaxation of the route based formulation in general, does not produce strong lower bounds.

Another important possible extension of our algorithm concerns the two-echelon vehicle routing problem with time windows. This problem variant was considered by Dellaert et al. (2019). However, the authors impose there a restrictive assumption that city freighter can receive products from only one urban truck. It would be useful to consider the 2E-CVRP with time windows without this assumption.

Acknowledgments

We are grateful to François Vanderbeck whose efforts contributed to obtaining the Ph.D. studentship for the first author. We also want to thank Artur Alves Pessoa who inspired the idea of the modified formulation presented in the paper.

Experiments presented in this paper were carried out using the PlaFRIM (Federative Platform for Research in Computer Science and Mathematics), created under the Inria PlaFRIM development action with support from Bordeaux INP, LABRI and IMB and other entities: Conseil Régional d'Aquitaine, Université de Bordeaux, CNRS and ANR in accordance to the "Programme d'Investissements d'Avenir".

References

Youcef Amarouche, Rym N. Guibadj, and Aziz Moukrim. A Neighborhood Search and Set Cover Hybrid Heuristic for the Two-Echelon Vehicle Routing Problem. In Ralf Borndörfer and Sabine Storandt, editors, *18th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2018)*, volume 65 of *OpenAccess Series in Informatics (OASIcs)*, pages 11:1–11:15, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Roberto Baldacci and Aristide Mingozzi. A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming*, 120(2):347–380, 2009.

Roberto Baldacci, Nicos Christofides, and Aristide Mingozzi. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, 115:351–385, 2008.

Roberto Baldacci, Aristide Mingozzi, and Roberto Roberti. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5):1269–1283, 2011.

Roberto Baldacci, Aristide Mingozzi, Roberto Roberti, and Roberto Wolfler Calvo. An exact algorithm for the two-echelon capacitated vehicle routing problem. *Operations Research*, 61 (2):298–314, 2013.

J. M. Belenguer, M. C. Martinez, and E. Mota. A lower bound for the split delivery vehicle routing problem. *Operations Research*, 48(5):801–810, 2000.

José-Manuel Belenguer, Enrique Benavent, Christian Prins, Caroline Prodhon, and Roberto Wolfler Calvo. A branch-and-cut method for the capacitated location-routing problem. *Computers & Operations Research*, 38(6):931 – 941, 2011.

U. Breunig, V. Schmid, R.F. Hartl, and T. Vidal. A large neighbourhood based heuristic for two-echelon routing problems. *Computers and Operations Research*, 76:208 – 225, 2016.

Claudio Contardo and Rafael Martinelli. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optimization*, 12:129 – 146, 2014.

Claudio Contardo, Vera Hemmelmayr, and Teodor Gabriel Crainic. Lower and upper bounds for the two-echelon capacitated location-routing problem. *Computers & Operations Research*, 39(12):3185 – 3199, 2012.

Claudio Contardo, Jean-François Cordeau Cordeau, and Bernard Gendron. A computational comparison of flow formulations for the capacitated location-routing problem. *Discrete Optimization*, 10(4):263 – 295, 2013.

Teodor Gabriel Crainic, Nicoletta Ricciardi, and Giovanni Storchi. Models for evaluating and planning city logistics systems. *Transportation science*, 43(4):432–454, 2009.

Teodor Gabriel Crainic, Guido Perboli, Simona Mancini, and Roberto Tadei. Two-echelon vehicle routing problem: A satellite location analysis. *Procedia - Social and Behavioral Sciences*, 2 (3):5944 – 5955, 2010.

Teodor Gabriel Crainic, Antonio Sforza, and Claudio Sterle. Location-routing models for two-echelon freight distribution system design. Technical Report 2011-40, CIRRELT, 2011.

R. Cuda, G. Guastaroba, and M.G. Speranza. A survey on two-echelon routing problems. *Computers and Operations Research*, 55:185 – 199, 2015.

Nico Dellaert, Fardin Dashty Saridarq, Tom Van Woensel, and Teodor Gabriel Crainic. Branch and price based algorithms for the two-echelon vehicle routing problem with time windows. *Transportation Science*, In Press, 2019.

I. Dunning, J. Huchette, and M. Lubin. JuMP: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.

Jesus Gonzalez-Feliu, Guido Perboli, Roberto Tadei, and Daniele Vigo. The two-echelon capacitated vehicle routing problem. Technical Report DEIS OR.INGCE 2007/2, Department of Electronics, Computer Science, and Systems, University of Bologna,, 2007.

Vera C. Hemmelmayr, Jean-François Cordeau, and Teodor Gabriel Crainic. An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & Operations Research*, 39(12):3215 – 3228, 2012.

Mads Jepsen, Bjorn Petersen, Simon Spoorendonk, and David Pisinger. Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, 56(2):497–511, 2008.

Mads Jepsen, Simon Spoorendonk, and Stefan Ropke. A branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle routing problem. *Transportation Science*, 47(1): 23–37, 2013.

G. Laporte and Y. Nobert. A branch and bound algorithm for the capacitated vehicle routing problem. *Operations-Research-Spektrum*, 5(2):77–85, Jun 1983.

Adam N. Letchford and Juan-José Salazar-González. Projection results for vehicle routing. *Mathematical Programming*, 105(2):251–274, Feb 2006.

Jens Lysgaard. CVRPSEP: a package of separation routines for the capacitated vehicle routing problem, 2018. URL `http://econ.au.dk/research/researcher-websites/jens-lysgaard/cvrpsep/`.

Jens Lysgaard, Adam N. Letchford, and Richard W. Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2):423–445, Jun 2004.

Diego Pecin, Artur Pessoa, Marcus Poggi, and Eduardo Uchoa. Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation*, 9(1):61–100, 2017a.

Diego Pecin, Artur Pessoa, Marcus Poggi, Eduardo Uchoa, and Haroldo Santos. Limited memory rank-1 cuts for vehicle routing problems. *Operations Research Letters*, 45(3):206 – 209, 2017b.

Guido Perboli, Roberto Tadei, and Daniele Vigo. The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. *Transportation Science*, 45(3):364–380, 2011.

Artur Pessoa, Marcus Poggi de Aragão, Marcus, and Eduardo Uchoa. Robust branch-cut-and-price algorithms for vehicle routing problems. In Bruce Golden, S. Raghavan, and Edward Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43 of *Operations Research/Computer Science Interfaces*, pages 297–325. Springer US, 2008.

Artur Pessoa, Eduardo Uchoa, and Marcus Poggi de Aragão. A robust branch-cut-and-price algorithm for the heterogeneous fleet vehicle routing problem. *Networks*, 54(4):167–177, 2009.

Artur Pessoa, Ruslan Sadykov, Eduardo Uchoa, and François Vanderbeck. Automation and combination of linear-programming based stabilization techniques in column generation. *INFORMS Journal on Computing*, 30(2):339–360, 2018.

Artur Pessoa, Ruslan Sadykov, Eduardo Uchoa, and François Vanderbeck. A generic exact solver for vehicle routing and related problems. In Andrea Lodi and Viswanath Nagarajan, editors, *Integer Programming and Combinatorial Optimization*, volume 11480 of *Lecture Notes in Computer Science*, pages 354–369, Cham, 2019. Springer International Publishing.

Edward Rothberg. An evolutionary algorithm for polishing mixed integer programming solutions. *INFORMS Journal on Computing*, 19(4):534–541, 2007.

Ruslan Sadykov, Eduardo Uchoa, and Artur Pessoa. A bucket graph based labeling algorithm with application to vehicle routing. Cadernos do LOGIS 7, Universidade Federal Fluminense, October 2017.

Ruslan Sadykov, François Vanderbeck, Artur Pessoa, Issam Tahiri, and Eduardo Uchoa. Primal heuristics for branch-and-price: the assets of diving methods. *INFORMS Journal on Computing*, In Press, 2018.

Fernando Afonso Santos, Geraldo Robson Mateus, and Alexandre Salles da Cunha. A branch-and-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. *Transportation Science*, 49(2):355–368, 2015.

Michael Schneider and Maximilian Löffler. Large composite neighborhoods for the capacitated location-routing problem. *Transportation Science*, 53(1):301–318, 2019.

Eiichi Taniguchi and Russell G Thompson. Modeling city logistics. *Transportation research record*, 1790(1):45–51, 2002.

François Vanderbeck, Ruslan Sadykov, and Issam Tahiri. BaPCod — a generic Branch-And-Price Code, 2019. URL `https://realopt.bordeaux.inria.fr/?page_id=2`.

Kangzhou Wang, Yeming Shao, and Weihua Zhou. Matheuristic for a two-echelon capacitated vehicle routing problem with environmental considerations in city logistics service. *Transportation Research Part D: Transport and Environment*, 57:262 – 276, 2017.

Zheng-Yang Zeng, Wei-Sheng Xu, Zhi-Yu Xu, and Wei-Hui Shao. A hybrid GRASP+VND heuristic for the two-echelon vehicle routing problem arising in city logistics,. *Mathematical Problems in Engineering*, pages 1–11, 2014.

A Detailed BCP results for literature instances

In following tables, Rg stands for the root gap, Rt stands for the time spent in the root node, BPB is the best primal bound found, and t stands for the total time spent. Note that if the time in the column "Literature" has prefix ">" for a given instance, Baldacci et al. (2013) did not solve this instance to optimality.

In sets 4A and 4B, instances `Instance50-`i`.dat` with $i = 1, 2 \ldots 18$ have two satellites. Instances with $i = 19, 20 \ldots 36$ have three satellites. Instances with $i = 37, 38 \ldots 54$ have five satellites. In set 5, instance names have the format `2eVRP_`i`_`j`_`k`.dat` with i the number of customers, j the number of satellites, and k an identifier. In sets $6A$ and $6B$, instance names have the form I`-n`i`-`j`.dat` with an identifier I that is the letter A, B or C, $i-1$ the number of customers, and j the number of satellites.

Table 5: Results of experiments on instances of set 4A

	Our best BCP with primal heuristic					Literature
Instance	Rg (%)	Rt (s)	Nodes	BPB	t (s)	t (s)
Instance50-1.dat	1.41	30	3	1569.42	58	47
Instance50-2.dat	0.00	35	1	1438.33	35	101
Instance50-3.dat	1.50	30	3	1570.43	67	44
Instance50-4.dat	0.20	55	3	1424.04	106	64
Instance50-5.dat	0.24	44	3	2193.52	92	415
Instance50-6.dat	0.00	55	1	1279.87	55	27
Instance50-7.dat	1.04	32	3	1458.63	76	63
Instance50-8.dat	0.33	58	3	1363.74	124	1414
Instance50-9.dat	0.94	32	3	1450.27	56	53
Instance50-10.dat	0.44	38	3	1407.65	77	71
Instance50-11.dat	0.53	29	15	2047.46	302	212
Instance50-12.dat	0.00	14	1	1209.42	14	43
Instance50-13.dat	1.36	35	3	1481.83	72	58
Instance50-14.dat	0.06	58	3	1393.61	93	743
Instance50-15.dat	1.47	39	3	1489.94	72	45
Instance50-16.dat	0.02	48	3	1389.17	84	39
Instance50-17.dat	0.17	41	3	2088.49	98	191
Instance50-18.dat	0.65	47	3	1227.61	90	73
Instance50-19.dat	0.68	33	3	1564.66	56	146
Instance50-20.dat	0.66	66	3	1272.97	111	88
Instance50-21.dat	0.53	38	3	1577.82	67	137
Instance50-22.dat	0.00	55	1	1281.83	55	50
Instance50-23.dat	0.92	41	5	1807.35	88	944
Instance50-24.dat	0.00	49	1	1282.68	49	50
Instance50-25.dat	0.61	40	3	1522.42	80	210
Instance50-26.dat	0.12	49	3	1167.46	73	34
Instance50-27.dat	0.43	61	3	1481.57	115	222
Instance50-28.dat	0.73	68	3	1210.44	148	185
Instance50-29.dat	1.21	38	13	1722.04	287	>5683
Instance50-30.dat	1.30	87	3	1211.59	333	152
Instance50-31.dat	1.98	53	5	1490.33	226	>7226
Instance50-32.dat	1.31	73	3	1199.00	205	2506
Instance50-33.dat	1.37	55	3	1508.30	116	>8076
Instance50-34.dat	0.93	69	5	1233.92	307	129
Instance50-35.dat	1.55	42	5	1718.41	859	>12736
Instance50-36.dat	0.56	92	3	1228.89	196	96
Instance50-37.dat	1.62	77	7	1528.73	328	505
Instance50-38.dat	1.51	151	3	1169.20	409	1030
Instance50-39.dat	1.23	58	3	1520.92	119	434
Instance50-40.dat	2.02	83	3	1199.42	245	623
Instance50-41.dat	1.34	57	7	1667.96	375	840
Instance50-42.dat	0.27	133	3	1194.54	367	140
Instance50-43.dat	1.41	64	9	1439.67	376	685
Instance50-44.dat	1.62	99	3	1045.12	322	272
Instance50-45.dat	0.72	85	3	1450.96	154	484
Instance50-46.dat	1.57	68	7	1088.77	350	841
Instance50-47.dat	0.65	90	5	1587.29	274	979
Instance50-48.dat	0.07	58	3	1082.20	90	57
Instance50-49.dat	1.16	77	3	1434.88	181	447

Instance	Our best BCP with primal heuristic					Literature
	Rg (%)	Rt (s)	Nodes	BPB	t (s)	t (s)
Instance50-50.dat	2.11	117	5	1083.12	595	836
Instance50-51.dat	0.89	87	3	1398.05	144	468
Instance50-52.dat	2.60	94	5	1125.67	485	959
Instance50-53.dat	1.44	81	7	1567.77	1557	2640
Instance50-54.dat	1.80	83	3	1127.61	346	651

Table 6: Results of experiments on instances of set 4B

Instance	Our best BCP with primal heuristic					Literature
	Rg (%)	Rt (s)	Nodes	BPB	t (s)	t (s)
Instance50-1.dat	1.61	40	5	1569.42	126	73
Instance50-2.dat	0.00	42	1	1438.33	42	118
Instance50-3.dat	1.46	41	5	1570.43	249	61
Instance50-4.dat	0.20	80	3	1424.04	146	72
Instance50-5.dat	0.25	41	3	2193.52	87	395
Instance50-6.dat	0.00	62	1	1279.87	62	33
Instance50-7.dat	0.09	36	3	1408.57	61	48
Instance50-8.dat	0.27	61	3	1360.32	117	2058
Instance50-9.dat	0.00	38	1	1403.53	38	51
Instance50-10.dat	0.00	47	1	1360.56	47	29
Instance50-11.dat	0.53	27	17	2047.46	445	282
Instance50-12.dat	2.37	89	5	1209.42	459	70
Instance50-13.dat	0.59	56	3	1450.93	321	59
Instance50-14.dat	0.04	63	3	1393.61	105	669
Instance50-15.dat	1.25	47	3	1466.83	168	66
Instance50-16.dat	0.44	59	3	1387.83	110	62
Instance50-17.dat	0.18	42	3	2088.49	86	224
Instance50-18.dat	1.16	52	5	1227.61	368	80
Instance50-19.dat	2.03	46	11	1546.28	474	183
Instance50-20.dat	0.67	65	3	1272.97	126	107
Instance50-21.dat	0.97	37	3	1577.82	78	157
Instance50-22.dat	0.00	70	1	1281.83	70	43
Instance50-23.dat	1.07	61	3	1652.98	122	121
Instance50-24.dat	0.00	33	1	1282.68	33	50
Instance50-25.dat	0.06	47	3	1408.57	89	97
Instance50-26.dat	0.03	68	3	1167.46	96	35
Instance50-27.dat	0.37	62	3	1444.50	122	124
Instance50-28.dat	1.99	81	5	1210.44	369	156
Instance50-29.dat	0.62	76	3	1552.66	123	162
Instance50-30.dat	2.34	94	5	1211.59	470	154
Instance50-31.dat	1.40	55	5	1440.86	257	164
Instance50-32.dat	1.36	79	5	1199.00	313	2383
Instance50-33.dat	1.14	73	7	1478.86	252	292
Instance50-34.dat	1.00	79	5	1233.92	363	180
Instance50-35.dat	2.34	88	5	1570.72	516	812
Instance50-36.dat	0.84	74	3	1228.89	186	113
Instance50-37.dat	2.03	91	19	1528.73	1184	>9076
Instance50-38.dat	0.63	161	5	1163.07	715	727
Instance50-39.dat	2.71	71	9	1520.92	822	1119
Instance50-40.dat	1.40	156	3	1163.04	630	218
Instance50-41.dat	1.26	97	3	1652.98	235	1482
Instance50-42.dat	0.71	120	3	1190.17	345	270
Instance50-43.dat	0.79	75	5	1406.11	235	6936
Instance50-44.dat	0.90	82	5	1035.03	353	242
Instance50-45.dat	1.81	67	7	1401.87	653	303
Instance50-46.dat	1.98	110	5	1058.11	339	267
Instance50-47.dat	0.82	96	3	1552.66	169	767
Instance50-48.dat	0.28	78	3	1074.50	142	78
Instance50-49.dat	1.21	91	5	1434.88	342	>8713
Instance50-50.dat	0.49	126	3	1065.25	225	318
Instance50-51.dat	2.22	111	5	1387.51	497	812
Instance50-52.dat	2.29	160	5	1103.42	1060	529
Instance50-53.dat	1.51	89	5	1545.73	273	1497
Instance50-54.dat	1.23	105	3	1113.62	234	642

Table 7: Results of experiments on instances of set 5

	Our best BCP with primal heuristic					Literature
Instance	Rg (%)	Rt (s)	Nodes	BPB	t (s)	t (s)
2eVRP_100-5-1b.dat	1.97	121	443	1104.06*	35993	>15018
2eVRP_100-5-1.dat	0.89	59	189	1564.46	5331	5850
2eVRP_100-5-2b.dat	1.71	125	167	782.25	10643	>16312
2eVRP_100-5-2.dat	1.96	63	365	1016.32	8416	6574
2eVRP_100-5-3b.dat	1.14	147	5	828.54	531	>20434
2eVRP_100-5-3.dat	0.37	65	3	1045.29	95	1831
2eVRP_100-10-1b.dat	3.00	335	71	911.80	12139	–
2eVRP_100-10-1.dat	1.35	201	7	1124.93	621	–
2eVRP_100-10-2b.dat	0.73	311	7	766.28	1040	–
2eVRP_100-10-2.dat	0.66	133	7	985.40	346	–
2eVRP_100-10-3b.dat	2.34	351	71	848.16	16894	–
2eVRP_100-10-3.dat	0.54	243	3	1042.63	451	–
2eVRP_200-10-1b.dat	2.23	1653	19	1173.07*	35958	–
2eVRP_200-10-1.dat	1.06	633	11	1537.52	2804	–
2eVRP_200-10-2b.dat	0.81	1345	7	985.99	4132	–
2eVRP_200-10-2.dat	0.45	555	9	1352.87	1658	–
2eVRP_200-10-3b.dat	3.78	1611	27	1221.42*	36336	–
2eVRP_200-10-3.dat	0.35	775	15	1777.49	2988	–

* These best primal bounds have not been proved optimal. The gap between the best dual bound and the best primal bound is 1.54% for 2eVRP_100-5-1.dat, 0.53% for 2eVRP_200-10-1b.dat, and 3.13% for 2eVRP_200-10-3b.dat.

Table 8: Results of experiments on instances of set 6A

	Our best BCP with primal heuristic					Literature
Instance	Rg (%)	Rt (s)	Nodes	BPB	t (s)	t (s)
A-n51-4.dat	0.87	58	3	652.00	115	74
A-n51-5.dat	0.39	74	3	663.41	125	97
A-n51-6.dat	0.13	98	3	662.51	148	164
A-n76-4.dat	0.78	51	3	985.95	94	215
A-n76-5.dat	1.50	66	5	979.15	148	536
A-n76-6.dat	1.82	83	9	970.20	368	2080
A-n101-4.dat	1.25	91	5	1194.17	278	3732
A-n101-5.dat	0.63	104	5	1211.35	481	3015
A-n101-6.dat	1.74	131	27	1155.89	1483	>73798
B-n51-4.dat	0.31	45	3	563.98	80	36
B-n51-5.dat	0.03	72	3	549.23	118	82
B-n51-6.dat	0.00	64	1	556.32	64	78
B-n76-4.dat	0.36	49	3	792.73	77	209
B-n76-5.dat	0.59	60	3	783.93	95	382
B-n76-6.dat	0.54	78	3	774.17	133	1297
B-n101-4.dat	0.29	93	3	939.21	165	1570
B-n101-5.dat	0.68	130	5	967.82	330	4412
B-n101-6.dat	0.63	115	5	960.29	242	2358
C-n51-4.dat	0.26	54	3	689.18	112	49
C-n51-5.dat	0.94	83	3	723.12	181	270
C-n51-6.dat	0.62	87	3	697.00	182	104
C-n76-4.dat	0.92	50	3	1054.89	142	284
C-n76-5.dat	3.16	79	15	1115.32	775	1136
C-n76-6.dat	2.92	104	73	1060.52	3840	>29901
C-n101-4.dat	0.65	92	5	1292.04	304	>18516
C-n101-5.dat	0.71	119	5	1304.86	446	>6791
C-n101-6.dat	1.23	144	47	1284.48	2235	>17481

Table 9: Results of experiments on instances of set 6B

	Our best BCP with primal heuristic					Literature
Instance	Rg (%)	Rt (s)	Nodes	BPB	t (s)	t (s)
A-n51-4.dat	0.05	50	3	744.24	95	55
A-n51-5.dat	0.42	48	3	811.51	69	54
A-n51-6.dat	0.63	69	3	930.11	140	240
A-n76-4.dat	0.65	62	3	1385.51	125	416
A-n76-5.dat	0.57	59	3	1519.86	106	311
A-n76-6.dat	0.30	69	3	1666.06	127	430
A-n101-4.dat	0.79	97	9	1881.44	589	2100
A-n101-5.dat	0.45	124	15	1709.06	1033	>6928
A-n101-6.dat	1.02	119	11	1777.69	886	>23948
B-n51-4.dat	0.00	58	1	653.09	58	20
B-n51-5.dat	0.02	38	3	672.10	60	27
B-n51-6.dat	0.48	67	3	767.13	122	77
B-n76-4.dat	0.46	43	3	1094.52	81	79
B-n76-5.dat	0.14	54	3	1218.12	96	82
B-n76-6.dat	0.17	71	3	1326.76	114	143
B-n101-4.dat	0.43	85	5	1499.71	232	>3673
B-n101-5.dat	0.96	117	11	1394.79	729	>20189
B-n101-6.dat	0.44	121	5	1445.97	300	>3970
C-n51-4.dat	0.22	63	3	866.58	108	75
C-n51-5.dat	0.74	72	3	943.12	138	149
C-n51-6.dat	0.49	76	3	1050.42	126	182
C-n76-4.dat	0.22	59	3	1438.96	94	114
C-n76-5.dat	0.79	68	5	1745.39	234	525
C-n76-6.dat	0.34	78	3	1756.46	139	902
C-n101-4.dat	0.57	76	25	2064.86	1900	>7550
C-n101-5.dat	0.50	126	11	1962.52	687	>7762
C-n101-6.dat	0.62	140	7	1860.73	616	>14553

B Detailed results for unsolved instances using a special BCP parameterisation

We have run our BCP algorithm for three unsolved instances 2eVRP_100-5-1b.dat, 2eVRP_200-10-1b.dat, and 2eVRP_200-10-3b.dat using a special parameterisation for each instance, and initial primal bounds which are greater than the best known solution values. All instances were solved to optimality within 60 hours. Table 10 provides the results for this three instances.

Table 10: Results for three instances unsolved with our best BCP in 10 hours

	BCP with special parameterisation					
Instance	InitPB	Rg (%)	Rt (s)	Nodes	BPB	t (s)
2eVRP_100-5-1b.dat	1104.00	1.39	166	809	1099.35	66212
2eVRP_200-10-1b.dat	1177.00	1.69	2634	47	1173.07	28667
2eVRP_200-10-3b.dat	1201.00	0.98	3288	159	1192.35	181652

C Detailed BCP results for new instances

In set 7, instance names have the format `2e-`i`-`j`-`k`.dat` with i the number of customers, j the number of satellites, and k an identifier.

Table 11: Results of experiments on instances of set 7

	Our best BCP with primal heuristic					
Instance	Rg (%)	Rt (s)	Nodes	BLB	BPB	t (s)
2e-100-5-1c.dat	0.40	78	3	1284.59	1284.59	121
2e-100-5-2c.dat	0.49	68	3	821.42	821.42	98
2e-100-5-3c.dat	0.00	67	1	841.17	841.17	67
2e-100-5-4a.dat	2.56	53	11	895.37	895.37	339
2e-100-5-4b.dat	5.34	174	5	560.25	560.25	530
2e-100-10-1c.dat	0.52	221	5	961.61	961.61	401
2e-100-10-2c.dat	0.48	145	3	860.66	860.66	217
2e-100-10-3c.dat	0.48	105	3	815.32	815.32	155
2e-100-10-4a.dat	2.17	261	629	886.61	886.61	34096
2e-100-10-4b.dat	1.15	436	11	594.70	594.70	2458
2e-200-10-1c.dat	1.28	726	63	1513.95	1513.95	10180
2e-200-10-2c.dat	1.06	885	259	1370.65	1370.65	25385
2e-200-10-3c.dat	0.58	1206	415	1793.82	1793.82	85502
2e-200-10-4a.dat	1.23	1213	221	1411.80	1411.80	46575
2e-200-10-4b.dat	2.56	1960	19	896.96	910.23	217533
2e-200-15-1a.dat	2.01	3319	89	1512.42	1535.11	215920
2e-200-15-1b.dat	3.00	5165	99	982.56	1001.43	216404
2e-200-15-1c.dat	2.07	2822	203	1439.06	1461.80	215915
2e-200-15-2a.dat	0.42	1562	13	1493.41	1493.41	8153
2e-200-15-2b.dat	1.13	5461	5	916.78	916.78	22446
2e-200-15-2c.dat	0.72	2777	5	1275.75	1275.75	14201
2e-200-15-3a.dat	0.81	2988	35	1569.77	1569.77	50171
2e-200-15-3b.dat	0.57	3688	19	972.28	972.28	49405
2e-200-15-3c.dat	1.96	3528	209	1313.21	1330.52	216018
2e-200-15-4a.dat	4.40	4187	87	1317.25	1366.52	216174
2e-200-15-4b.dat	∞	5391	13	859.95	∞	215936
2e-200-15-4c.dat	1.79	3159	319	1386.61	1403.5	215908
2e-300-10-1a.dat	0.75	3289	281	4223.34	4223.34	164537
2e-300-10-1b.dat	2.96	3913	103	2541.05	2596.69	215964
2e-300-10-1c.dat	3.59	2770	73	4781.32	4920.97	215908
2e-300-10-2a.dat	0.79	2420	381	4040.08	4060.08	215888
2e-300-10-2b.dat	∞	2576	37	2286.88	∞	215907
2e-300-10-2c.dat	2.38	1989	115	3546.18	3613.03	215897
2e-300-10-3a.dat	0.94	2654	189	4008.59	4008.59	86489
2e-300-10-3b.dat	∞	4677	75	2315.30	∞	216040
2e-300-10-3c.dat	∞	2253	123	4590.85	∞	215907
2e-300-10-4a.dat	0.24	1846	11	4094.94	4094.94	4219
2e-300-10-4b.dat	∞	2851	73	2339.32	∞	215907
2e-300-10-4c.dat	0.45	2495	225	3938.17	3938.17	59323
2e-300-15-1a.dat	3.20	6796	47	3948.09	4058.67	215828
2e-300-15-1b.dat	∞	12431	45	2460.70	∞	216062
2e-300-15-1c.dat	2.42	6659	47	4135.19	4219.51	216136
2e-300-15-2a.dat	3.18	6177	49	3591.91	3671.50	215823
2e-300-15-2b.dat	–	–	–	–	–	–
2e-300-15-2c.dat	∞	5902	57	3497.77	∞	215991
2e-300-15-3a.dat	3.98	6459	51	3410.43	3522.31	216034
2e-300-15-3b.dat	3.39	8563	19	2113.98	2175.68	215959
2e-300-15-3c.dat	–	–	–	–	–	–
2e-300-15-4a.dat	∞	5571	47	3738.63	∞	215869
2e-300-15-4b.dat	∞	10773	23	2173.11	∞	215818
2e-300-15-4c.dat	1.30	6976	201	3575.95	3600.79	215821

We cannot provide any result for instances 2e-300-15-2b.dat and 2e-300-15-3c.dat because the LP solver could not solve the LP relaxation of the restricted master during the first column generation convergence.